AIAA 2005-6835

Evolutionary Computing for Low-Thrust Navigation

Seungwon Lee,* Wolfgang Fink,† Ryan P. Russell,‡
Paul von Allmen,§ Anastassios E. Petropoulos,** and Richard J Terrile††
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

The development of new mission concepts requires efficient methodologies to analyze, design, and simulate the concepts before implementation. New mission concepts are increasingly considering the use of ion thrusters for fuel-efficient navigation in deep space. This paper presents parallel, evolutionary computing methods to design trajectories of spacecraft propelled by ion thrusters and assesses the trade-off between delivered payload mass and required flight time. The developed methods utilize a distributed computing environment in order to speed up computation, and use evolutionary algorithms to find globally Pareto-optimal solutions. The methods are coupled with two main traditional trajectory design approaches, which are called direct and indirect. In the direct approach, thrust control is discretized in either arc time or arc length, and the resulting discrete thrust vectors are optimized. In the indirect approach, the thrust control problem is transformed into a co-state control problem and the initial values of the co-state vector are optimized. The developed methods are applied to two problems: 1) an orbit transfer around the Earth and 2) a transfer between two distance retrograde orbits around Europa, the icy Galilean moon closest to Jupiter. The optimal solutions found with the present methods are comparable to other state-of-the-art trajectory optimizers, while the required computation time is often several orders of magnitude less thanks to an intelligent design of control vector discretization, advanced algorithmic parameterization, and parallel computing.

Nomenclature

a	=	semimajor axis
e	=	eccentricity
i	=	inclination
ω	=	argument of the periapsis
Ω	=	longitude of the ascending node
$\boldsymbol{u}$	=	control vector
$\boldsymbol{x}$	=	state vector
$\boldsymbol{\lambda}$	=	co-state vector
t	=	time
dt	=	time step
f, p	=	generic functions
J	=	performance index
ψ	=	boundary condition

* Member of the Information and Computer Science Staff at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.
† Senior Researcher at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.
‡ Member of the Engineering Staff at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.
§ Senior Member of the Information and Computer Science Staff at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.
** Senior Member of the Engineering Staff at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.
†† Senior Member of the Technical Staff at JPL, 4800 Oak Grove Dr. Pasadena, CA, 91109.

I. Introduction

THIS paper addresses the problem of finding optimal orbit transfers for low-thrust spacecraft. A common goal for the optimization problem is to find the minimum-time, minimum-fuel, or Pareto-optimal trajectories, where the Pareto-optimality means that no other solutions are superior to them in terms of both flight time and fuel consumption. Particularly in an early mission design phase, where maximizing the deliverable payload mass is an equally attractive mission objective as minimizing time of flight, the Pareto-optimal solutions that demonstrate the trade-off between flight time and deliverable payload mass are desired. In general, these optimization problems are difficult to solve not only due to the long transfer time and multi-revolutionary transfer but also due to the search for Pareto-optimality.

Various methods have been used to solve this optimization problem. A majority of the work has utilized either direct or indirect methods.[1] The direct method approaches the problem by adjusting the control variables iteratively to reduce a performance index such as flight time and propellant mass.[2,3] The continuous control and state variables are often discretized, which results in a nonlinear programming. The indirect method, on the other hand, makes use of the control law that arises when the low-thrust spacecraft problem is formulated using calculus of variations.[4] A co-state vector is introduced and the thrust history is completely determined by the initial values of the co-states. These initial values become the optimization parameters and the only remaining constraint is to hit a specified terminal condition.

For the nonlinear programming problems in the direct method and the indirect method, local gradient-based algorithms such as Newton's method and sequential quadratic programming are popular because they are widely available and proven very effective for many applications.[5] However, the traditional algorithms find locally optimal solutions typically in the vicinity of the initial guesses. Additionally, the algorithms are unstable when the objective function is rugged and the function gradient is discontinuous. The trajectory optimization problem tends to have many locally optimal solutions, which makes it difficult to find the globally optimal solution. Furthermore, the traditional optimization algorithms do not directly handle the multi-objective problem but convert it into a single scalar objective, the so-called weighting method. The resulting solution is highly sensitive to the weighting factors and is a single solution rather than a set of Pareto-optimal solutions. Therefore, the low-thrust orbit transfer problem calls for a more robust, global, and Pareto-optimal optimization algorithm.

This paper presents innovative methods to solve the nonlinear programming problems of the direct and indirect methods for the low-thrust orbit transfer optimization. The present method consists of two global-search algorithms: a genetic algorithm and simulated annealing.[6–10] Neither algorithms require the objective function gradient and are likely to find a globally optimal solution in a rugged search domain, as opposed to gradient-based algorithms. Additionally, the genetic algorithm takes advantage of a population-based search to directly solve the multi-objective optimization problem in a single run. Moreover, the simulated annealing algorithm exploits a highly non-local ensemble search namely "shotgun" mode (described in later section) to efficiently solve the problem in a single parallel run, which exhibits a perfect linear speed-up on a cluster computer due to no communication overhead.

This paper is an overview of an ongoing research program and highlights the key features of the methods and results. The technical details of the approaches are left for future papers.

II. Methodologies

The following section defines the problem of the low-thrust trajectory optimization in a general way and describes the present method developed for the optimization problem.

A. Low-Thrust Trajectory Optimization Problem

Electric propulsion systems are one of the most efficient propulsion systems available for space missions. At the expense of longer flight times, these systems use less propellant and thereby require a less massive spacecraft and a smaller launch vehicle. This propellant efficiency makes the electric propulsion system attractive for a budget-sensitive space program. The thrust provided by the electric propulsion system is relatively small, typically on the order of one Newton. As a result, any significant maneuver of a spacecraft with the electric propulsion system requires continuous thrust over long periods of time. This makes the low-thrust trajectory optimization more challenging than the chemical-propulsion spacecraft trajectory optimization where only a few impulsive maneuvers need to be optimized.

The equations of motion for a spacecraft are given by a set of ordinary differential equations, which include the effect of the engine thrust, gravitational sources, and inertial forces, if non-inertial reference systems are used.

$$\dot{\boldsymbol{x}}(t) = f\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right], \tag{1}$$

where $\boldsymbol{x(t)}$ is the state vector consisting of the position, velocity, and mass of the spacecraft, $\boldsymbol{u(t)}$ is the control vector representing the thrust acceleration vector, and t is time. The low-thrust orbit transfer optimization problem is to minimize the performance index *J,* subject to the equations of motion and the initial and final boundary conditions ψ_o (initial) and ψ_f (final):

$$J = \int_{t_o}^{t_f} p\left[\boldsymbol{x}(t),\boldsymbol{u}(t),t\right]dt$$

$$\text{subject to} \tag{2}$$

$$\psi\left[\boldsymbol{x}(t_o),\boldsymbol{u}(t_o),t_o\right] = \psi_o \text{ and } \psi\left[\boldsymbol{x}(t_f),\boldsymbol{u}(t_f),t_f\right] = \psi_f .$$

Typically, the performance index J is given by the spacecraft final mass, the time of flight, or the linear combination of the two quantities. An alternative approach is to use a Pareto-dominance concept to simultaneously optimize both the final mass and the flight time. The resulting solutions are Pareto-optimal solutions that are considered equally good in terms of the vector objective given by the final mass and the flight time. Both approaches (scalar and vector objectives) are implemented in this paper and described in a later section.

B. Direct Method

The direct method "directly" optimizes the thrust control variables $\boldsymbol{u(t)}$ to minimize/maximize a performance index. In this method, the continuous control variables are discretized in either arc time or arc length. The discrete control variables are directly adjusted and optimized to satisfy the boundary conditions and optimization criteria. There are various ways to discretize the control variables, and the choice of the discretization strategy can strongly affect the performance of the optimization. In general, a finer discretization leads to a better approximation of the original control problem, but requires significantly more computation time. Several discretization strategies are implemented and tested to help guide an optimal strategy for a given problem.

C. Indirect Method

The indirect method introduces a co-state vector and "indirectly" optimizes the thrust control variables by adjusting the initial values of the co-state vector. This method uses the optimality conditions that arise from calculus of variations to transform the control vector to a function of the initial values of the co-states only. Traditionally, this transformation combined with the Euler-Lagrange boundary conditions, or the so-called transversality conditions, leads to a two-point boundary-value problem, whose solution inherently satisfies the conditions for local optimality.[4] In the current approach, the transversality conditions are ignored, and the initial values of the co-states are iterated to directly optimize the desired objective. This hybrid method is termed indirect however, because the co-states are the control parameters in lieu of the thrust vector itself. The approach as stated is well suited for integration into any general constrained optimization framework, such as the current multi-objective evolutionary computing approaches. This approach is particularly attractive because it avoids one of the problems typically associated with indirect methods: i.e. gradients and transversality conditions must be tediously re-derived each time when an objective or constraint is changed. The transversality conditions are indirectly satisfied by optimizing the initial co-states, and the gradients are not required for the proposed evolutionary optimization methods. Therefore, the problem can be tailored for custom applications with relative ease compared to traditional methods.

D. Global Optimization Method

In order to solve the global optimization problems, which appear in the nonlinear programming problem of the direct and indirect methods, the present method uses a genetic algorithm and simulated annealing. The genetic algorithm is inspired by the natural selection and sexual reproduction process of living organisms,[6–8] and the simulated annealing mimics the thermodynamic process of cooling molten metals.[9,10] Both methods have mechanisms to escape from local minima in order to find a globally optimal solution. The global search mechanism is a reproduction operator with a stochastic selection mechanism in the case of the genetic algorithm and a mutation operator with the Metropolis algorithm in the case of the simulated annealing. For the genetic algorithm, the following parameters are typically used: the population size of 1000, the maximum number of generations of 100, the crossover probability of 0.8, the mutation rate per gene of 1/N, where N is the number of genes, the elitist fraction of 30%.

E. Pareto-Optimization Method

For the Pareto-optimization, the genetic algorithm handles directly multiple objectives with non-dominated sorting in a single run.[11,12] The non-dominated sorting uses the concepts of non-dominance and dominance to rank the population composed of candidate solutions.[13] When comparing two solutions, a solution is termed dominated if the solution is inferior to the other solution in all objectives. Otherwise, the solution is termed non-dominated. The non-dominated sorting finds solutions that are non-dominated in comparison with the rest of the candidate solutions in the population. The non-dominated solutions constitute a first Pareto-front and are assigned the best fitness value. The sorting continues with the dominated solutions (i.e., the complement of the non-dominated solutions) by finding the next Pareto-front and assigning a slightly worse fitness value. Since the non-dominated sorting does not involve a weighting process of aggregating the multi-objectives into a single scalar objective function, a careful, educated initial guess of the weighting factors is not needed. The genetic algorithm accompanied by the non-dominated sorting can generate Pareto-optimal solutions in a single synergetic optimization run. The concept of non-dominated sorting does not apply to the simulated annealing approach used here.

F. Constraint Handling Method

The low-thrust trajectory optimization problems involve not only multiple objectives but also multiple constraints such as the boundary condition for a spacecraft final state to meet a given target state. Typically, the constraints are treated with a penalty function as part of the fitness/energy function.[14] The penalty function approach requires a weighting process when combining the penalty function and the objective function into a single scalar fitness function. This approach is used for the simulated annealing application. A different approach named stochastic ranking is used in the genetic algorithm application.[15] The stochastic ranking method strikes a balance between the objectives and the constraints in their contributions to the population ranking process by randomly choosing the ranking criterion between the two. A user-defined parameter determines how probable it is to choose one criterion versus the other.

G. "Shotgun" Mode in Simulated Annealing

For the simulated annealing, a "shotgun" approach is applied to improve the computational efficiency. In this mode, we start with a random set of values for the co-state parameters and expose them to a fixed (user-defined) number of iterations while the "temperature" of the annealing process is oscillating. If, within the specified iterations, the target orbit is reached with a user-defined accuracy based on the current set of parameters that prescribe an optimal trajectory, then a "solution" is found and stored. Otherwise, the algorithm starts over with another set of random initial values for parameters to optimize. In the results presented here, the number of simulated annealing iterations used for the indirect method optimization was 100.

H. Parallel Computing Method

The genetic algorithm uses a population-based search and thus is amenable to parallel computing. When the fitness evaluation is one of the computationally more expensive parts, the parallel computing becomes an ideal choice to reduce the computation time. The fitness evaluation of the candidate solutions in the population is distributed among several processors in the distributed memory system. The evaluation result is sent to the master processor on which the rest of the algorithmic process such as parent selection, offspring creation, and population replacement is executed. The fitness-value passing is the only message passing between the processors in the genetic algorithm run. As a result, the computational overhead due to the parallel computing is marginal.

For the simulated annealing, an "embarrassingly" parallel approach is taken for parallel computing. A perfect linear speed up is demonstrated since each processor performs an independent optimization run without cross-communication (message passing) between processors. We have demonstrated a 1002 CPU run on JPL's institutional cluster computer (Cosmos, 37th largest super computer in the world to date).

III. Results

The present method is applied to two types of trajectory problems: A) two-body orbit transfer problem and B) restricted three-body orbit transfer problem. The optimization results are presented and compared with solutions found with other state-of-the-art optimizers in terms of solution optimality and computational efficiency.

A. Orbit Transfer around the Earth

As an example of two-body orbit transfer problems, a low-thrust orbit transfer around the Earth, is considered. The Earth is the only gravitational body in this problem and is approximated as a point mass. The optimization problem is to find Pareto-optimal solutions for the transfer from a low-eccentricity, small orbit to a high-eccentricity, coplanar, larger orbit. The initial orbit has a semimajor axis of 9,222.7 km and an eccentricity of 0.2, and the final orbit has a semimajor axis of 30,000 km and an eccentricity of 0.7. A relatively high thrust magnitude of 9.3 N is used for this transfer problem. The specific impulse of the thrust engine is set to 3,100 s. The initial mass of the spacecraft is 300 kg.

Within the direct method, several discretization strategies for the approximation of the continuous control vectors are tested. One strategy is to discretize the control vectors into uniform-time thrust arcs. Another strategy is to discretize the control vectors into uniform-angle thrust arcs. Both strategies lead to an increasing number of discrete control vectors as the flight time increases, and become quickly impractical for multi-revolutionary transfers. To overcome this problem, a simplified strategy is developed for the coplanar transfer. The strategy allocates two thrust arcs per revolution, with the arcs being centered at each of the two apsides. The length of the thrust arc and the direction of the thrust vector are the resulting variables to optimize. Ideally, the thrust-arc length and the thrust-vector direction should vary from one revolution to another. However, the same length and vector is used for every revolution in order to reduce the number of independent variables to optimize. Although this strategy leads to only four independent parameters (two thrust-arc lengths and two thrust-vector angles), it captures energetically economic maneuvers, which occur near the apsides in the coplanar transfer. It should be noted that this is not a generalizable approach but is expected to be sub-optimal for this type of orbit-transfer.

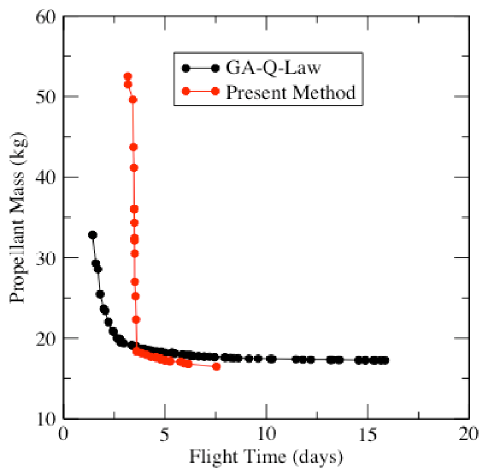

Figure 1. Pareto-optimal solutions found with the present method and GA-Q-Law for the low-thrust orbit transfer around the Earth.

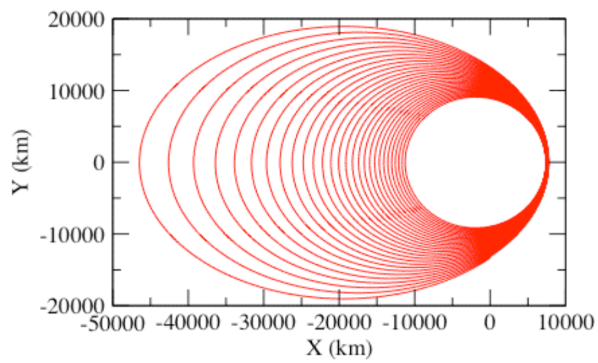

Figure 2. Minimum-fuel trajectory found with the present method for the low-thrust orbit transfer around the Earth.

Figure 1 shows the Pareto-optimal solutions found with the present method. The present solutions are compared with the solutions found with GA-Q-Law,[16] which is an optimized heuristic control law based on a Lyapunov feedback control law named Q-law,[17] optimized by a genetic algorithm. It has been demonstrated that the GA-Q-Law finds nearly Pareto-optimal solutions in a reasonable computation time.[16] The present solutions are as good as the GA-Q-Law solutions for a long flight time case, while they are not as good as the GA-Q-Law solutions for a short flight time case (below 7 days). This comparison shows both the shortcoming and effectiveness of the simplified discretization strategy of the present method. It is expected that the present strategy would be inefficient for a short flight time case where a continuously varying control vector is essential to improve the fuel efficiency. Figure 2 shows the minimum-fuel trajectory among the Pareto-optimal solutions found with the present method.

B. Distant Retrograde Orbit (DRO) Transfer around Europa

As a restricted three-body orbit transfer problem, a DRO transfer around Europa, the icy Galilean moon closest to Jupiter, is considered. The gravitational fields of Jupiter and Europa are included as point masses while the gravitational fields of the other moons are excluded. The dynamics of the spacecraft is described in the rotating frame where Europa is at the center, the x-axis points along the Jupiter-Europa line, and the z-axis points along Europa's angular momentum vector with respect to Jupiter. The initial DRO is given by the position vector (0.07518, 0) and the velocity vector (0, -0.14992) in the x-y plane with the unit length of 67,0988 km and the unit time of

48831.6 seconds. Similarly, the final DRO has the position vector (0.03067, 0) and the velocity vector (0, -0.07274). The spacecraft is modeled with the specific impulse of 7,365 s, the thrust magnitude of 4.984 N, and the initial mass of 25,000 kg, which is a typical spacecraft mass for JIMO (Jupiter Icy Moon Orbiter) Mission.

Both the direct and indirect methods are applied to this DRO transfer optimization. In the direct method, the same discretization strategy used for the orbit transfer around the Earth is chosen since it is found to capture the energetically efficient maneuvers of this DRO transfer as well. The two thrust angles (each at one of the apsides) and the two thrust-arc lengths are optimized to meet the boundary condition as well as to maximize performance indices. The objectives considered are the flight time and ΔV, which is the integration over time of the magnitude of the acceleration produced by the propulsion engine. In the indirect method, the initial values of the co-state vector associated with a thrust angle, a thrust switching function, and their time derivatives are optimized to minimize the flight time and ΔV, while satisfying the final-state boundary conditions given by the target DRO.

Figure 3 shows the Pareto-optimal solutions found with the direct and indirect methods. The present optimization results are compared with the solutions found by Mystic, which is a high-fidelity trajectory optimization software package based on the static/dynamic control algorithm developed by Whiffen.[18,19] The direct method with the two constant thrust arcs per revolution does not perform as well as the other optimization methods. However, the indirect method yields results that are comparable to assumed-optimal Mystic solutions. The comparison demonstrates that both the genetic algorithm and simulated annealing efficiently solve the constrained multi-objective problem and find Pareto-optimal solutions.

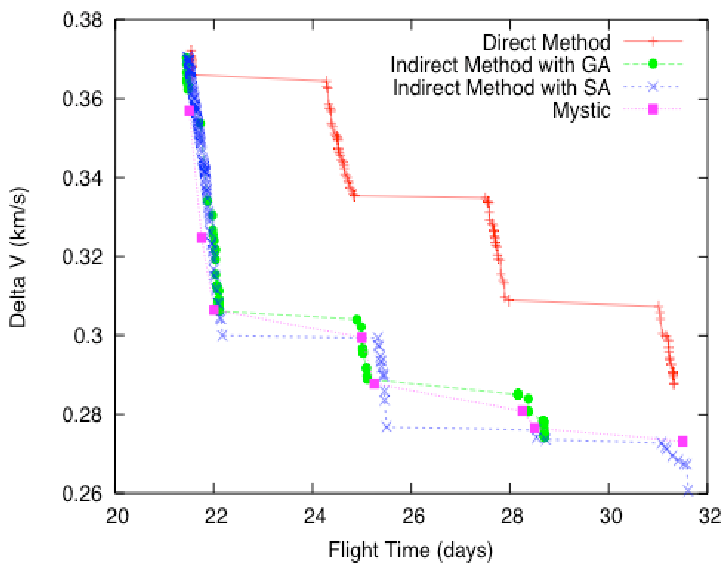

Figure 3. Pareto-optimal solutions found with the direct and indirect methods for the DRO transfer around Europa.

In terms of domain expertise requirements, the present optimization method requires considerably less than Mystic. Mystic requires an educated initial guess to start its optimization process.[20] Without the educated guess, the computation time may increase by up to one order of magnitude or may fail to converge entirely.[20] The Mystic solutions shown in Figure 3 are obtained with a good initial guess based on a heuristic control law, the Q-law.[17] In contrast, the present method automatically conducts an optimization process with minimal guidance from a domain expert for a reasonable bound for each variable. Furthermore, the computational efficiency and solution optimality are less sensitive to the quality of the inputs in the present method compared to Mystic.

The computation time used for the Pareto-optimal solutions compares as follows: With Intel 3.2 GHz Xeon processors, Mystic used about 300 minutes on one processor, while the present method used about 5 minutes in wall-clock time on 16 processors for the genetic algorithm and 300 minutes on 256 processors for the simulated annealing (this can be modified to lower CPU numbers or shorter runs).

One main difference between Mystic and the present method is that several independent runs are required to obtain the Pareto-optimal solutions with Mystic while the present method (both genetic algorithm and simulated annealing) generates the Pareto-optimal solutions in a single synergetic run. Mystic leads to one optimal solution per run by increasing a target flight time incrementally for each run. Therefore, Mystic cannot directly exploit cooperative interactions among Pareto-optimal solutions during the optimization process. It can only indirectly take advantage of a previously found solution by using it as an initial guess for the next optimization run with a slightly different flight time. However, this dependency between runs prevents Mystic from having several runs in parallel, and as a result leads to long sequential computation runs.

Figure 4 shows the variations of the trajectory and control profile for Pareto-optimal solutions. The top row illustrates the trajectories with flight times of 21.4 days, 21.7 days, and 22.0 days from left to right. The middle row depicts the trajectories with flight times of 24.9 days, 25.0 days, and 25.1 days, and the bottom row represents the trajectories with flight times of 28.2 days, 28.4 days, and 28.7 days. The minimum-flight-time trajectory (the top-left figure in Fig. 4) shows a continuous thrust arc (red line). As the flight time increases, several coast arcs (green dashed lines) are inserted around the *y* axis. This observation suggests that the present strategy taken for the control vector discretization in the direct method is a reasonable approximation, although it is not as general and efficient as the indirect method.

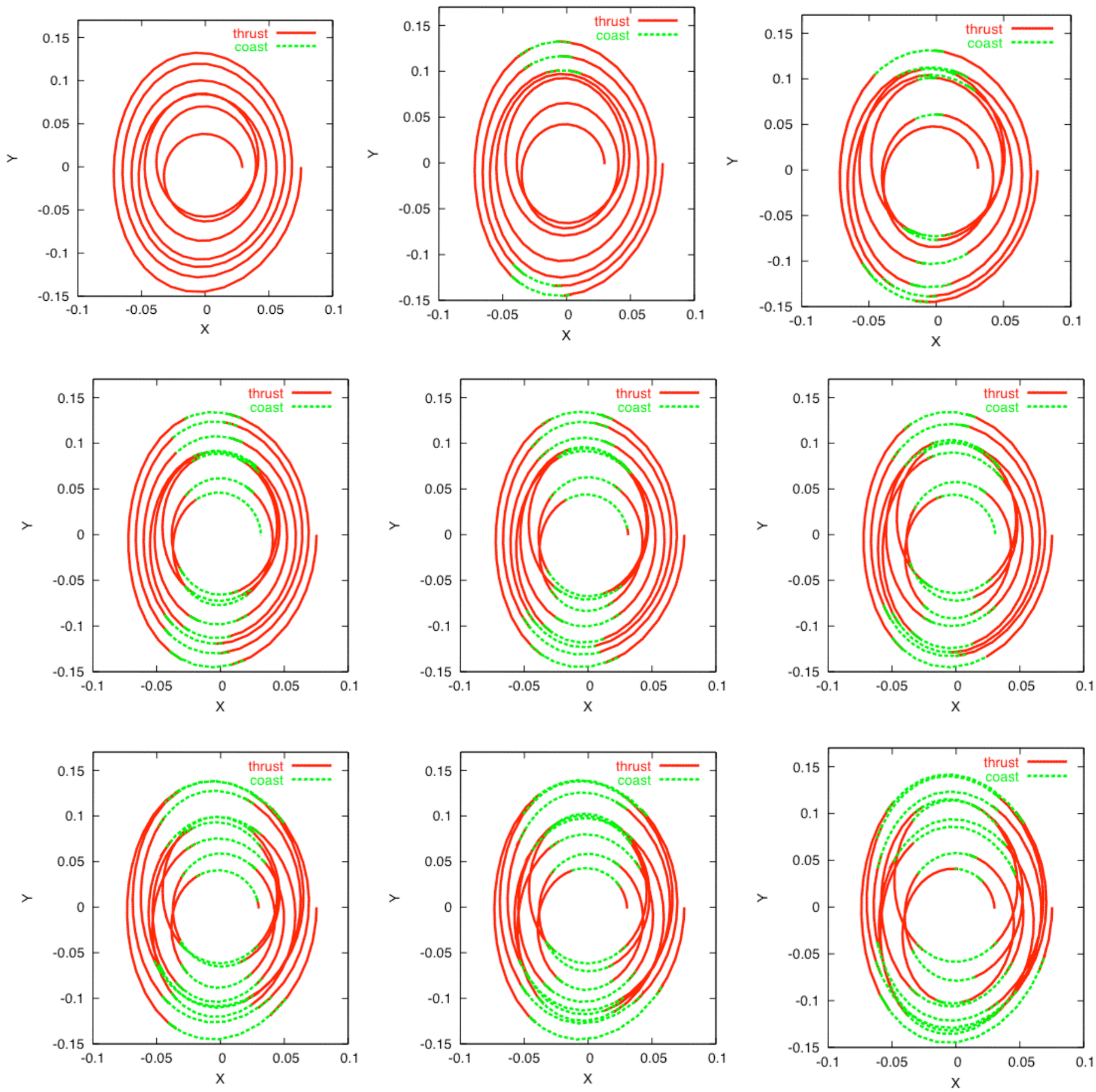

Figure 4. Pareto-optimal trajectories found with the indirect method optimization. From the top left panel to the bottom right, the flight time of the trajectory increases, and more and longer coast arcs appear.

IV. Conclusions

We have developed a robust and efficient method for the global Pareto-optimization of low-thrust orbit transfers by applying a genetic algorithm and simulated annealing to both the direct and indirect methods. The direct method "directly" optimizes the thrust control variables to reduce a performance index such as flight time and propellant mass, while the indirect method introduces a co-state vector and "indirectly" optimizes the thrust control variables by adjusting the initial values of the co-states. Both direct and indirect methods lead to a constrained, multi-objective optimization problem, which the present method solves with the two global-search algorithms. The investigated example problems demonstrate that this method finds nearly Pareto-optimal solutions with a high computational efficiency and minimal guidance from domain expertise. The high computational efficiency is obtained by taking advantage of an intelligent design of control vector discretization, advanced algorithmic parameterization, and parallel and synergetic computing. Future applications of the present method include the optimization of more complex two-body orbit transfers and restricted three-body orbit transfers, which appear frequently in space missions around Earth and beyond.

Acknowledgments

We thank Christoph Adami for useful discussions about performance analysis for genetic algorithms, and we thank Bob Tisdale for his support on the parallelization of the simulated annealing. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The research was supported by JPL's Research and Technology Development program.

References

[1]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193-207.

[2]Kluever, C. A. and Oleson, S. R., "A Direct Approach for Computing Near-Optimal Low-Thrust Transfers," *AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 97-717, Sun Valley, Idaho, Aug. 4-7, 1997.

[3]McConaghy, T. T. and Longuski, J. M., "Parameterization Effects on Convergence when Optimizing a Low-Thrust Trajectory with Gravity Assists," AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA Paper 2005-5403, Providence, Rhode Island, Aug. 16-19, 2004.

[4]Jezewski, D. J., "Primer Vector Theory and Applications," Technical Report NASA TR R-454, Lyndon B. Johnson Space Center, Houston, Texas, 77058, November 1975.

[5]Fletcher, R., *Practical Methods of Optimization*, John Willey & Sons, 2 edition, 2000.

[6]Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.

[7]Mitchell, M., *An Introduction to Genetic Algorithms (Complex Adaptive Systems),* The MIT Press, 1998.

[8]Eiben, A. E. and Smith, J. E., *Introduction to Evolutionary Computing*, Springer, 2003.

[9]N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, "Equation of State Calculation by Fast Computing Machines*," J. of Chem. Phys.*, **21**, 1087--1091, 1953.

[10]S. Kirkpatrick, C.D. Gelat, M.P. Vecchi,, "Optimization by Simulated Annealing," *Science*, **220**, 671--680, 1983.

[11]Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[12]Coello, C. A. C., Van Veldhuizen, D. A., and Lamont, G. B., *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic Algorithms and Evolutionary Computation),* Penum US, 2002.

[13]Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, 2002, pp. 182-197.

[14]Fiacco, A. V. and McCormick, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, New York Wiley, 1968.

[15]Runarsson, T. P., and Yao, X., "Stochastic Ranking for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 3, 2000, pp. 284-294.

[16]Lee, S., von Allmen, P., Fink, W., Petropoulos, A. E., and Terrile, R. J., *"*Design and Optimization of Low-Thrust Orbit Transfers using the Q-law and Evolutionary Algorithms," *IEEE Aerospace Conference Proceedings*, Big Sky, Montana, Mar. 7-11, 2005.

[17]Petropoulos, A. E., "Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting," *AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 04-5089, Providence, Rhode Island, Aug. 16-19, 2004.

[18]Whiffen, G. J. and Sims, J. A., "Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization*,*" *AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 01-209, Santa Barbara, California, Feb. 11-15, 2001.

[19]Whiffen, G. J., "*Optimal Low-Thrust Orbit Transfers around a Rotating Non-Spherical Body*," *AAS/AIAA Space Flight Mechanics Meeting,* AAS Paper 04-264, Maui, Hawaii, Feb. 8-12, 2004.

[20]Petropoulos, A. E. and Lee, S., "Optimization of Low-Thrust Orbit Transfers Using the Q-Law for the Initial Guess," *AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 05-392, Lake Tahoe, Aug. 7-11, 2005.